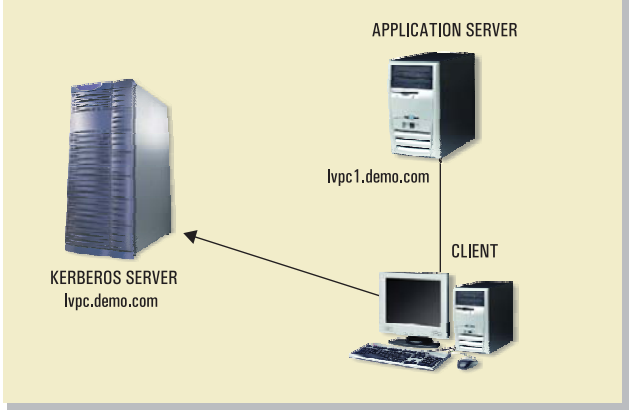
Installing Kerberos

Securing applications and protecting a system from network attacks is looking at winning only half the security battle. A computer system must also employ security policies at the protocol level, which Kerberos does very effectively. Learn how to install it.

In the first part of this series, we showed you how Kerberos can play a vital role in securing your network. In this second part, we will show you how you could set up Kerberos to secure your network.

Kerberos is a network authentication protocol from the Massachusetts Institute of Technology (MIT). It is designed to provide centralised strong authentication for client/server applications by using secret-key cryptography. It is a trusted third-party authentication service based on the model presented by Needham and Schroeder. (It is trusted in the sense that each client believes that the Kerberos judgment, with respect to the identity of the other client, is accurate.)

Figure 1: Components in the authentication process



There are three components involved in the authentication process in a given Kerberos realm: workstation, application server and the Kerberos server (refer Figure 1).

The workstation (or simply the client) is the one that runs Kerberised client applications like FTP client or Telnet client on behalf of the user. The application server (or simply the server) hosts the Kerberised server applications like FTP server, Telnet server, and so on. The Kerberos server hosts the Key Distribution Centre (KDC). The KDC is an integral part of the Kerberos system. It consists of three logical components: a database of all principals and their associated secret keys, the Authentication Server (AS) and the Ticket Granting Server (TGS). The AS issues Ticket Granting Ticket (TGT) to the clients who wish to use the services hosted by the application servers. The TGT can be used to request individual service tickets, while TGS issues individual service tickets to the clients.

In the rest of this article, we will show you the steps involved in setting up a Kerberos server (with hostname lvpc.demo.com), application server (with hostname lvpc1.demo.com) and the workstation. For the sake of simplicity, we will assume that all the three machines are running the same version of a Linux distribution and install the same version of the Kerberos package downloaded from the official MIT website. Only the system requirements and configurations make a difference.

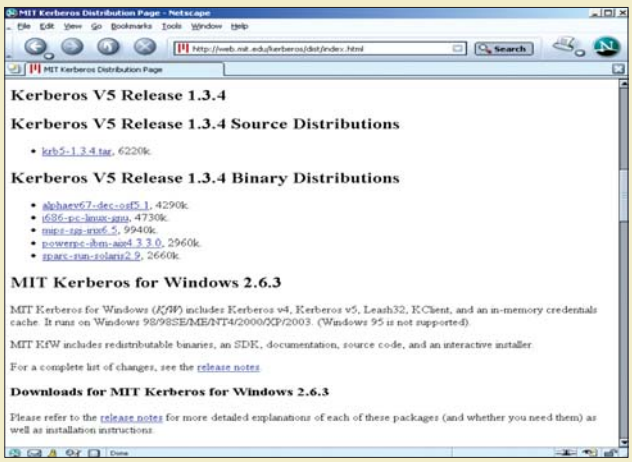


Figure 2: Official website of MIT

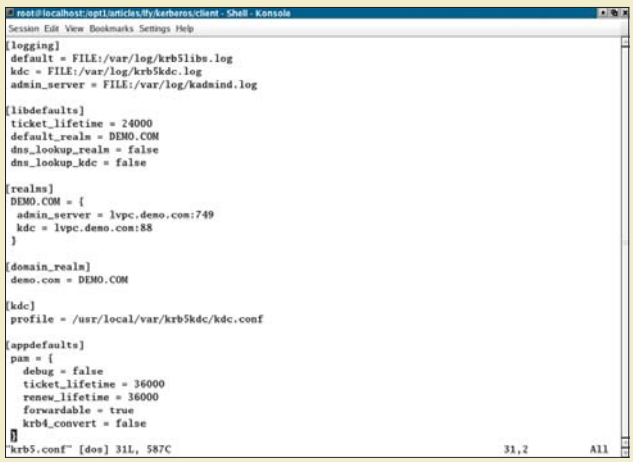


Figure 3: Default realm and the host of KDC

```
[kdcdefaults]
        kdc_ports = 88,749

[realms]
        DEMO.COM = {
                master_key_type = des-cbc-crc
                supported_enctypes = des3-cbc-sha1:normal
                        des3-cbc-sha1:norealm des3-cbc-sha1:onlyrealm
                        des-cbc-crc:v4 des-cbc-crc:afs3 des-cbc-crc:normal
                        des-cbc-crc:norealm des-cbc-crc:onlyrealm
                        des-cbc-md4:v4 des-cbc-md4:afs3 des-cbc-md4:normal
                        des-cbc-md4:norealm des-cbc-md4:onlyrealm
                        des-cbc-md5:v4 des-cbc-md5:afs3
                        des-cbc-md5:normal des-cbc-md5:norealm
                        des-cbc-md5:onlyrealm des-cbc-sha1:v4
                        des-cbc-sha1:afs3 des-cbc-sha1:normal
                        des-cbc-sha1:norealm des-cbc-sha1:onlyrealm
                acl_file = /usr/local/var/krb5kdc/kadm5.acl
                dict_file = /usr/share/dict/words
                admin_keytab = /etc/kadm5.keytab
                database_name = /usr/local/var/krb5kdc/principal
                key_stash_file = /usr/local/var/krb5kdc/.k5.DEMO.COM
                kadmind_port = 749
}
```

Figure 4: A sample kdc.conf file

Setting up the Kerberos server

The first step is to identify a server on which you will install the Kerberos KDC. The server you select should be:

- Physically secure.
- The operating system should be up to date with all the latest patches applied.
- There should be no user accounts on the machine, except for the Kerberos administrator.
- There should be as few processes as possible running on the server, other than the Kerberos daemons.
- The system clock should be synchronised with the Network Time Protocol (NTP).
- It should be capable of resolving names with the DNS server.

Once you have identified the server, the next step is to get the latest source or binary version of Kerberos from the official MIT website (refer Figure 2).

If you download the binary version, all that you may need to do is to install or extract the downloaded file. In case you download the source version for some reason, you may need to first build the Kerberos binaries as described in the document http://web.mit.edu/kerberos/www/krb5-1.3/krb5-1.3.3/doc/krb5-install.html#Building%20Kerberos%20V5.

The next step is to set up the *krb5.conf* and *kdc.conf* files. The *krb5.conf* file contains Kerberos configuration information. Normally, you should install your *krb5.conf* file in the directory */etc*. A sample *krb5.conf* is shown in Figure 3.

Figure 3 also shows that the default realm is DEMO.COM and the KDC is running on the host lvpc.demo.com listening at port 88. The *kadmin* server is running on the same host as the KDC and listening on the port 749. The location of the *kdc.conf* file is */usr/local/var/krb5kdc/kdc.conf*.

The *kdc.conf* file (refer Figure 4) contains Kerberos KDC configuration information. Normally, you should install your *kdc.conf* file either in the directory */usr/local/var/krb5kdc* or in the location specified in the KDC section of *krb5.conf*.

After setting up the *krb5.conf* and *kdc.conf* files, you need to create the Kerberos database using the *kdb5_util* like this...

```
#kdb5_util create -s
Loading random database
Initialising database '/usr/local/var/krb5kdc/principal' for
realm 'DEMO.COM',
Master key name 'K/M@DEMO.COM'

You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
```

```
Re-enter KDC database master key to verify:

#
```

These commands will create the database files under */usr /local/var/krb5kdc* directory as specified in the *database_name* property in the realm section of *kdc.conf*.

The next step is to create the Access Control List (ACL) file and add the Kerberos administrator principal into it. A sample ACL entry is...

```
root@DEMO.COM *
```

The location of the ACL file should match the *acl_file* property in the realm section of *kdc.conf*.

To add client principals to the Kerberos KDC you may first need to add the administrator principal (the one you added to

ACL file) to the Kerberos database using the *kadmin.local* command like this...

```
#kadmin.local -p root@DEMO.COM
Authenticating as principal root@DEMO.COM with password
kadmin.local: addprinc root@DEMO.COM
WARNING: no policy specified for root@DEMO.COM; defaulting to no
policy
Enter password for principal "root@DEMO.COM":
Re-enter password for principal "root@DEMO.COM":
Principal "root@DEMO.COM" created.
Kadmin.local: listprincs
K/M@DEMO.COM
kadmin/admin@DEMO.COM
kadmin/changepw@DEMO.COM
kadmin/history@DEMO.COM
krbtgt/DEMO.COM@DEMO.COM
root@DEMO.COM
kadmin.local: quit
#
```

After adding the administrator principal you can go ahead and add other user principals in the same way as you did for the administrator principal.

For each client host (AS) to access the Kerberos server you need to add the host principal to the Kerberos database like this...

```
#kadmin.local -p root@DEMO.COM
Authenticating as principal root@DEMO.COM with password
kadmin.local: addprinc -randkey host/ivpc1.demo.com
WARNING: no policy specified for host/1vpc.demo.com@DEMO.COM,
defaulting
to no policy
Principal "host/1vpc.demo.com@DEMO.COM" created
kadmin.local: quit
#
```

As a last step you need to start the Kerberos KDC and Kadmind servers like this...

```
/usr/local/sbin/krb5kdc
/usr/local/sbin/kadmind
```

Setting up the application server

The Kerberos package that you downloaded from the MIT website will have all the required Kerberised server programs like *ftpd, telnetd* and so on. So go ahead and install that package on the AS machine. You can find the Kerberised server programs in the */usr/local/sbin* directory. For the system to use the Kerberised version of the server programs add */usr/local/sbin* to the system path. In addition to that, if your system uses *xinetd* to handle the services, then you need to update the *ftpd* file in */etc/xinet.d* to use the Kerberised *ftpd* from */usr/local/sbin* like this...

```
Service ftp
{
        disable = no
        socket_type = stream
        wait = no
        user = root
        server = /usr/local/sbin/ftpd
        nice
}
```

The next step is to set up the *keytab* file. */etc/krb5.keytab* is required for the AS to authenticate the Kerberos server. Actually, the *keytab* file is a local encrypted copy of the host's secret key. You have to remember that if this file is compromised, then the entire Kerberos system could be compromised. So make sure that *keytab* is readable only to the root user. Other users have no job with the *keytab* file. The *keytab* file can be either generated on the Kerberos server and transferred to the AS through some secure mechanism like sftp etc, or it can be generated on the AS itself by using the *kadmin* service like this...

```
#kadmin -p root@DEMO.COM
kadmin5: ktadd host/1vpc1.demo.com

Entry for principal host/1vpc1.demo.com with kvno3, encryption
type Triple DES cbc mode with HMAC/sha 1 added to keytab WRFILE:
etc/krb5:keytab.
Entry for principal host/1vpc.demo.com with kvno 3, encryption
type DES cbc mode with CRC-32 added to keytab WRFILE:/etc/
krb5.keytab

kadmin5: quit
#
```

You have to remember that you may be able to generate the *keytab* file for the AS only because you added this AS host to the Kerberos database while setting up the Kerberos server.

And the final thing that you may need to do is to configure the */etc/krb5.conf* file. You can use the same */etc/krb5.conf* file as you used for the Kerberos server.

Setting up the workstation

Workstation installation is much more straightforward and easier than the installations of AS and the Kerberos server.

Since all the required Kerberised client programs are shipped with the Kerberos package downloaded from the MIT website, all that you may need to do is to install that package on the workstation machine. You can see the Kerberised client programs installed in the directory */usr/local/bin* and */usr/local/sbin*.

For the users to use the Kerberised version of the client programs, like ftp, Telnet, etc, add the */usr/local/bin* and */usr/local/sbin* directories ahead of */bin* and */usr/bin* in the system path.

Also, you need to configure the */etc/krb5.conf* file. You can use the same */etc/krb5.conf* file as you used for the Kerberos server and AS.

Using Kerberos

The first step to access a service running on the AS is to get the TGT from the TGS running on the Kerberos server like this...

```
#kinit root@DEMO.COM
Password for root@DEMO.COM:
#
```

The TGT will permit you to get additional tickets for specific services like ftp, Telnet, and so on.

If you happen to use the Kerberised version of the login program to login to your system, then you would have got the TGT automatically.

To view the tickets that were stored in the ticket cache, use the *klist* command like this...

```
#klist
Ticket cache: Error! Hyperlink reference not valid.
Default principal: root@DEMO.COM
Valid starting Expires Service principal
05/27/04 11:03:44 05/28/04 11:02:34 krbtgt/DEMO.COM@DEMO.COM
Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
#
```

In this example, the ticket cache file */tmp/krb5cc_0* is the file in which all the tickets are stored. The default principal is root@DEMO.COM for which the KDC has issued the TGT. The 'Valid starting' and 'Expires' field denotes the ticket lifetime. The Service principal describes each ticket. The TGT has the primary 'krbtgt' and the instance 'DEMO.COM' which is nothing but the realm name.

To access a service running on the AS using Kerberos authentication, you need to use the Kerberised version of the client programs that you have installed on the workstation machine like this...

```
# /usr/local/bin/ftp 1vpc1
Connected to 1vpc1.demo.com
220 1vpc1.demo.com FTP server (Version 5.60) ready
334 Using authentication type GSSAPI; ADAT must follow
GSS API accepted as authentication type
GSS API authentication succeeded
Name (1vpc1 :root): root
232 GSS API user root@DEMO.COM is authorised as root
Remote system type is Unix
Using binary mode to transfer files
ftp>
ftp> bye
221 Goodbye
#

# klist
Ticket cache: FILE: /tmp/krb5cc_0
Default principal: root@DEMO.COM

Valid starting Expires Service principal
05/27/04 11:03:44 05/28/04 11:02:34 krbtgt/DEMO.COM@DEMO.COM
05/27/04 11:03:44 05/28/04 11:02:34 host/1vpc1.demo.com@DEMO.COM

Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
#
```

The *klist* example shows you the Kerberised FTP client connected to Kerberised FTP server using the GSSAPI (generic security service application program interface). The client has got the service ticket 'host/lvpc1.demo.com@DEMO.COM' automatically to access the service. You should note that the user root has not entered her password to access the service. This clearly shows that no password transfer over an insecure network is required in the Kerberos authentication mechanism.

Once you have done with using the service, you can destroy the tickets stored in the ticket cache using the *kdestroy* program like this...

```
# kdestroy
# klist
klist: No credentials cache found (ticket cache: FILE: /tmp/
krb5cc_0)
Kerberos 4 ticket cache: /tmp/tkt0
klist: You have no tickets cached
```

If you want to change the password you need to use the *kpasswd* instead of *passwd* like this...

```
# kpasswd root@DEMO.COM
Password for root@DEMO.COM:
Enter new password::
Enter it again::
Password changed.
```

That's all you need to do to set up Kerberos to secure your network! Pretty simple, isn't it? LFY

By: Raja R K. The author is a lead engineer working with HCL Technologies (Cisco Systems Offshore Development Centre) in Chennai. He can be contacted at: rajark_hcl@yahoo.co.in